



**UNIVERSIDAD DE
GUADALAJARA**

**CENTRO
UNIVERSITARIO DE
CIENCIAS EXACTAS
E INGENIERÍAS**



**DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN
CIBER-HUMANA**

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Valdivia Guerra Diego Alberto-214546944

Seminario de Traductores de lenguaje II

2024-A

Introducción

La combinación de un analizador léxico para generar tokens y un analizador sintáctico basado en tablas LR para realizar el análisis sintáctico permite procesar y comprender la estructura del código fuente de un programa, facilitando así su posterior traducción o interpretación en acciones concretas. Este enfoque es fundamental en el desarrollo de compiladores y entornos de desarrollo integrados (IDE) para diversos lenguajes de programación.

Desarrollo

```
tests > C prueba en C.c
1  int main(){
2      int a,b,c;
3      a = 10;
4      b = 20;
5      c = 30;
6      if((a + b) == c){
7          return c;
8      }
9  }
```

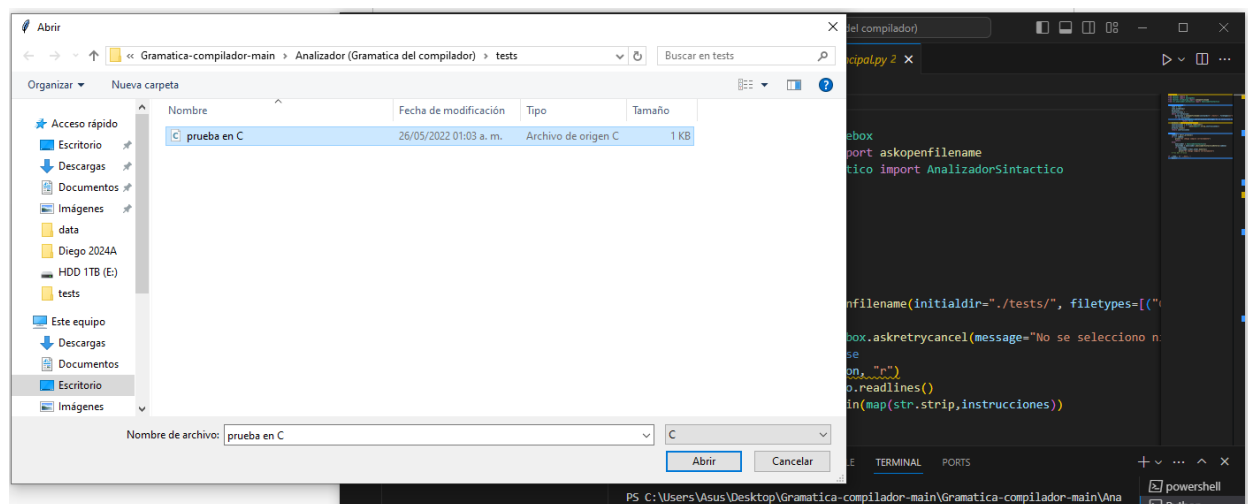
Con un archivo donde se realizara la prueba y que tomara el código como referencia para hacer el análisis léxico

[illegible]

Tomando como referencia la tabla en Excel proporcionada, para tomar las reglas al momento de hacer el análisis

```
data > compilador.lr
1 24 1 programa
2 25 0 Definiciones
3 25 2 Definiciones
4 26 1 Definicion
5 26 1 Definicion
6 27 4 DefVar
7 28 0 ListaVar
8 28 3 ListaVar
9 29 6 DefFunc
10 30 0 Parametros
11 30 3 Parametros
12 31 0 ListaParam
13 31 4 ListaParam
14 32 3 BloqFunc
15 33 0 DefLocales
16 33 2 DefLocales
17 34 1 DefLocal
18 34 1 DefLocal
19 35 0 Sentencias
20 35 2 Sentencias
21 36 4 SentenciaIdent
22 36 6 SentenciaIf
23 36 5 SentenciaWhile
24 36 3 SentenciaReturn
25 36 2 SentenciaLlamada
26 37 0 Otro
27 37 2 Otro
28 38 3 Bloque
29 39 0 ValorRegresa
30 39 1 ValorRegresa
31 40 0 Argumentos
32 40 2 Argumentos
33 41 0 ListaArgumentos
34 41 3 ListaArgumentos
35 42 4 SentenciaLlamada
```

Tenemos el archivo lr el cual se tomara en cuenta también formando parte del análisis léxico



Se nos pide hacer la apertura del archivo en este caso el .c el cual puede ser modificado para poder realizar el ejercicio

```
SentenciasReturn
ValorRegresa
ExpresiónTermino
TerminoIdent
C
Sentencias
SentenciasIf
DefLocales
Definiciones
El código compilo correctamente
PS C:\Users\Asus\Desktop\Gramatica-compilador-main\Gramatica-
```

```
analizador_Sintactico.py  prueba en C.c
tests > C prueba en C.c
1 int main(){
2     int a,b,c;
3     a = 10;
4     b = 20;
5     c = 30;
6     if((a + b) == c){
7         return c;
8     }
9 }
```

Se nos muestra en forma de lista las partes del código en secuencia

Conclusión

En resumen, estos conceptos y técnicas son fundamentales para la construcción de sistemas de software avanzados y contribuyen significativamente al desarrollo y la evolución de los lenguajes de programación y las tecnologías relacionadas.

Bibliografía

- Analizador lexico. Disponible en:
<http://materiacompiladores3110.blogspot.com/2017/02/analizador-lexico.html/>.
Consultado: 16 de marzo de 2020.
- SOFTWARE DE GENERACIÓN Y SIMULACIÓN DE TABLAS DE ANÁLISIS SINTÁCTICO (BURGRAM) - Construcción de las tablas de análisis LR. (n.d.).
http://cgosorio.es/BURGRAM/index75b8.html?Los_algoritmos_de_an%0Alisis%0Asint%0Actico:An%0Alisis_Sint%0Actico%0AAscendente:Construcci%0F3n_de_las_tablas_de%0Aan%0Alisis_LR